Traffic Forecasting Using Time Series Analysis

by

Mohammad Asifur Rahman Shuvo
16301132
Muhtadi Zubair
16301113
Sarowar Hossain
16301127
Afsara Tahsin Purnata
16301130

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
October 2020

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

ARShuvo

Mohammad Asifur Rahman Shuvo
16301132

Muhtadi

Muhtadi Zubair
16301113

Sarowar Hossain

Sarowar Hossain
16301127

Afsara Tahsin Purnata

Afsara Tahsin Purnata
16301130

Approval

The thesis titled "Traffic Forecasting Using Time-Series Analysis" submitted by

1. Mohammad Asifur Rahman Shuvo (16301132)
2. Muhtadi Zubair (16301113)
3. Sarowar Hossain (16301127)
4. Afsara Tahsin Purnata (16301130)

Of Summer, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 17, 2020.

Examining Committee:

Supervisor:

Dr. Muhammad Iqbal Hossain
Assistant professor
Department of computer science and engineering
BRAC University

Program Coordinator:

Md. Golam Rabiul Alam, PhD
Assistant Professor
Department of computer science and engineering
Brac University

Head of Department:

Prof. Mahbub Majumdar
Chairperson
Dept. of Computer Science & Engineering
Brac University

Mahbubul Alam Majumdar, PhD
Professor and Chairperson
Department of Computer Science and Engineering
Brac University

Ethics Statement

We, Mohammad Asifur Rahman Shuvo, Muhtadi Zubair, Sarowar Hossain, and Afsara Tahsin Purnata consciously assure that for the thesis paper "Traffic Forecast Using Time-Series Analysis" the following guidelines are followed:

1. This thesis done by us is our original work, which have not been published anywhere else previously.
2. Currently the paper is not being considered or thought about being published anywhere else.
3. Our analysis and research are being reflect by this paper in a complete and truthful manner.
4. All of our contributions are being properly credited by this paper.
5. In the context of existing and prior research, the results are appropriately placed.
6. Correct citations have been provided throughout the paper and sort of copying must be done by giving proper reference and using quotation marks.
7. We will publicly take responsibility for our paper's content as all of us has been actively involved with the writing of this paper.

We agree with the above policies and state that this submission follows the rules and regulation of BRAC University.

Date: 17/10/2020
Corresponding authors' signatures:

ARShuvo

Mohammad Asifur Rahman Shuvo
16301132

Muhtadi

Muhtadi Zubair
16301113

Sarowar Hossain

Sarowar Hossain
16301127

Afsara Tahsin Purnata

Afsara Tahsin Purnata
16301130

Abstract

Traffic jams are a common phenomenon all over the world, especially in a densely populated country like Bangladesh. Due to this, people try heart and soul to tackle this problem by any means necessary to save time to reach their desired destination. Hence the traffic related research is a hot topic now a days which will be quite beneficial for all people living in congested cities. We also tried to do some research on the traffic network to find the most suitable traffic forecasting model to forecast or predict the future traffic value using time-series forecasting models. The only topic which deals with both, traffic prediction and traffic control is traffic time-series analysis for which it is essential. In this paper, we have obtained a suitable data set containing data of the number of various vehicles for each hour for seven days straight. We have used this data set to feed into a few time-series forecasting models of our choosing. The models or algorithms considered are ARIMA, ETS, SNAIVE, PROPHET and the last one is the combination of all models we named it "mix". The study shows us the significant difference between each of the models and which one produces a more reliable and accurate prediction.

Keywords: Traffic Forecasting, Time-series forecasting models, ARIMA, SNAIVE, ETS, PROPHET

Acknowledgement

We cannot express our enough gratitude towards our Supervisor, Dr. Md. Iqbal Hossain, for his valuable time, support, help and encouragement.

Our Sir guided us throughout the whole project. It would be quite impossible to do an excellent research paper without our honorable supervisor's guidance. He had guided and supported us by all throughout our journey. Our supervisor has supported us by taking online sessions in this pandemic situation, which we appreciate from the bottom of our heart. All these have significantly helped us a lot to finish our fantastic thesis project work from home.

Nevertheless, we would want to show our appreciation to the Department of Computer Science and Engineering, BRAC University and our educators for all their help. It really meant a lot for us.

Table of Contents

List of Figures

List of Tables

List of Acronyms

The next list describes several symbols & abbreviation that will be later used within the body of the document

$ARIMA$ Auto-Regressive Integrated Moving Average

ETS Error Trend and Seasonality, or Exponential Smoothing

GAM Generalized Additive Model

MAE Mean Absolute Error

$MAPE$ Mean Absolute Percentage Error

$RMSE$ Root mean squared error

$SNAIVE$ Seasonal Naive

Chapter 1

Introduction

We all know that traffic jams are a significant issue for a big population country like Bangladesh. A traffic jam refers to a situation in a road when the bus, truck, rickshaw, and other vehicles are stuck for an extended period. Sometimes vehicles move very slowly because of the traffic jam. It mainly occurs in the morning when most of the people go to the office and students are going to school. On the other hand, garment workers go to their workplaces on foot walking by the side of the main road. It is a typical picture in the cities and towns and day by day it is increasing in cities and towns like Dhaka. In the present time, it is a big problem for all because most people do not reach their destination on time. One of the primary reasons for the traffic jam is overpopulation and increasing the number of vehicles day by day.

Moreover, there are many very narrow roads; that is why more than two vehicles are not moving side by side. There are many hawkers on the footpath, beside the main road, that is the reason for the traffic jam. Because hawkers block the footpath and people walk in the middle of the road and vehicles are not freely moving on the road. Sometimes, it is the reason for death while an ambulance is on its way to the hospital.

Commuters once controlled the traffic monitoring only on the streets of the main city, but in the present time, we can see that it is very easy to monitor the traffic in our hand's mobile. Traffic jams turn into a big issue day by day for us. Brac Institute of Government and Development states that the traffic problems in our capital city is affected by an estimate of five million or fifty lakhs working hours every day that may lead to costs of around 11.4 billion USD each year. Every day many people paid their extra money just because of the traffic because they want to go to their destination before any traffic on the road.

In the whole world, there are different countries that face traffic every day. However, our country has a large population. Every day many people are going to work and get stuck in the traffic badly. This is a preeminent problem for Dhaka city. However, the Government took some steps and built flyovers in the massive traffic area so that the traffic can reduce in the road. Most of the people do not maintain traffic rules, and they build up with severe traffic in the office time and in turn, severely affects the other people that are going to their destination. The economic growth of our total GDP is also impacted by this. Moreover, from Dhaka city, it can produce a

GDP of 36 per cent, and it made forty per cent of total employment of our country. [14]

1.1 Motivation

We decided to use time-series analysis for making Traffic forecasts and also choosing a model that will be quite beneficial for all people living in congested cities as it helps them get an overview of the traffic hours or a day before they plan to go to a particular place. The analysis of traffic time series is very important to predict the travel time and usually, it is also using a traffic control system. This is a system that operates with time-series data. The meaning of the time series is, in a fixed time of period data would generate in a series. Our aim is to the prediction of future traffic that will help us to know the traveling time. On the other hand, it impacts on our financial growth also. For example, if we forecast the electricity bill for the next month, then we will easily assume that how we manage our usage of electricity, it also helps us to save money. So that we want to decide that we will follow the time series analysis and forecasting, which fits our prediction model perfectly.

1.2 Objective

Traffic jam is a significant issue for our day to day life, but if we will know traffic for the next 24hours, then it is easy to set our working time as we are already faced with different types of problems during the traffic jam. For this purpose, we are working to find out the prediction of the next 24hours traffic. Prediction is a process where we assume the future value from the existing values. If we collect the current values for a particular road then we train that value after that, we can easily find out the next 24 hours prediction using time series analysis. We are trying to use four types of time series models, which are ARIMA, ETS, SNAIVE, and PROPHET.

1.3 Methodology

To implement our idea at first, we collected the historical data of traffic flow. Next, we applied some of the time-series analysis techniques and machine learning models to acquire our desired results. We have predicted the next 24 hours of traffic.

Time series has a lot of existing models such as ARIMA, VAR, Holt-Winters method, TAR, and so on. We decided to go with the ARIMA model because this is very common using for the forecasting data as well as while it is working it can also ignore the seasonality of the data. Traffic data is pretty similar throughout the year; hence no seasonality needs to be handled here. The full form of ARIMA is Auto-Regressive Integrated Moving Average. This is finding the forecast, analyze the past recorded data. Moreover, any non-seasonal time series also can exhibit the pattern with this model. For seasonal data SARIMA can perform also which is the short form of the Seasonal ARIMA. Other than ARIMA, the other models or algorithms considered are ETS, SNAIVE (Seasonal Naïve), PROPHET and the last one is the combination of all models we named it 'mix'. In this paper using which we were able to make a traffic forecasting model which can predict the number of vehicles particular date and time which currently limited to our dataset.

Chapter 2

Background

Time series analysis may be a statistical procedure that analyses and manipulates statistic data. It is made of collected data points at constant time intervals. Unlike regression analysis, time series analysis is instrumental in getting the critical statistics and characteristics of a time series data.[1]

As we know, a collection of data at regular intervals is time-series. Another technique that adds a bit more complexity but elegance to the system is forecasting. A technique that determines predictive estimates in determining future trends using historical data as input is called forecasting.[22] Now, this gets interesting when we combine this with time series. Now In Time series forecasting, the time sequence and data are analysed to predict future events. It is a machine learning technique. Here we use are using historical time series data as input.[6]

To understand time series analysis and time series forecasting with more clarity and depth, we looked into various published papers. The paper helped us with our thesis is discussed in the Literature Review section.

2.1 Literature Review

It is urgent for us for understanding the inner vehicle framework development rule, for expectation and control. To viably control gridlock and improve the traffic limit of the organization is important and critical for our general public. Hence, we have to comprehend that the Transportation framework is made out of various and discrete vehicle developments in the mind-boggling street organization and a shared limitation relationship exists between vehicle-to-vehicle which we have to do facilitate investigation.

In paper [17], a permeability chart calculation is utilized to change over traffic stream time arrangement into networks. Because of the changed over organizations, a few attributes will be found from the first traffic stream to recognize traffic states. To extensively examine traffic stream time arrangement in various thicknesses, two broad strategies have been presented as follows: multiracial detruded vacillation examination (MFDFA) and permeability diagram. They have taken around 3000 examples from an assigned street of just the left side which went about as their inspecting point. Their graphical information shows results dependent on 2000 of their examples. MFDFA and permeability diagrams are applied to find the intricacy and multifractality of time progress arrangement detached thickness. A few discoveries can be summed up as follows. Right off the bat, with the expectation of a complimentary stream, the time progress arrangement shows short-range connections and multifractality, while for a synchronized stream, the time progress arrangement displays long-range relationships and multifractality. Besides, a permeability chart is utilized to move fractal time arrangement into networks, and further geography boundaries have been dissected to show that the traffic organizing diverse thickness is described by the highlights of a without scale and little world.

In paper [8], they have examined how to group traffic time arrangements that have comparative vacillation designs. They have utilized a basic normal detrending strategy and just investigation the remaining time arrangement. Second, they have utilized standard investigation (PCA) on crude information and utilize the heaviness of the main d-parts as the highlights of the time arrangement. Third, they have utilized the k-implies calculation to group the traffic time arrangement. At long last, they examined the consequences of the bunching calculation and talked about the beginnings of the groups. All the information utilized in this paper is extricated from the openly available PMS dataset. The traffic stream time arrangement is recorded at 1000 road traffic stream stations from August 1, 2011, to August 31, 2011. The example timespan crude information is 5 minutes. In this paper, they have talked about the grouping of the traffic time arrangement that has comparative variance designs. It needs to take note of that a comparative change design isn't carefully identical to factual connection. They have discovered that the bunching results are exceptionally impacted by movement request designs.

In paper [11], they have made far-reaching learn about the key innovations including applying remote sensor organizations to the traffic observing organization, its traffic stream conjecture dependent on the dim determining model, and gridlock control. They additionally utilize the Adaptive GM (1,1) Model which makes some genuine memories moving estimate for city traffic and have a superior conjecture result. They have planned a calculation of traffic stream blockage control and booking for traffic organizations, which is called TRED. They have utilized it for continuous traffic planning and have opened up another approach to consider and take care of gridlock control issues.

Savvy traffic the board is one of the significant segments in keen city improvement as the vast majority of the metropolitan urban areas in India is confronting extreme gridlock because of expanding individual vehicle utilization [13]. Paper [13] presented an improved transient traffic stream determining calculation dependent on the ARIMA model. Forecast of traffic volume, i.e., the normal number of vehicles out and about in the following five minutes to one hour is a significant exploration issue in Intelligent Transportation Systems (ITS) applications. The normally utilized numerical procedures for predicting the traffic stream are the neural organization, relapse, chronicled normal calculations, Kalman sifting, and time arrangement examination. The examinations which depend on time arrangement investigation generally utilize the Box-Jenkins Autoregressive Integrated Moving normal (ARIMA) procedure. The issue with the ARIMA model is that it requires huge amounts of traffic volume information for the model turn of events and at times it is hard to comprehend and assess the model coefficients. The above-said requirements with ARIMA models spur the distinguishing proof of another procedure that might be utilized for the issue of traffic volume forecast utilizing just restricted information. Henceforth, in the current investigation [13], one of the old-style time arrangement strategies, specifically, multiplicative deterioration was attempted as an option in contrast to the ARIMA model for momentary traffic volume determining. The deterioration method is straightforward and easy to execute utilizing a spreadsheet or MATLAB program and it requires just restricted information.

To execute an Intelligent Transportation System successfully and proficiently the traffic stream information is fundamental. Such frameworks are created to help designs in reacting to gridlocks adequately and helpfully. This paper [2] proposed a model that predicts the traffic stream in blockages. One of the stars of this model is that it depends on the time arrangement. This model will have the option to anticipate the forthcoming traffic conditions in a specific crossing point. The creators guarantee that the model is equipped for anticipating the traffic stream with an exactness of 88.74% and 81.96% for 15minutes previously and 1hr before separately. They utilized 3 intelligent components, for example, states, data sources, and yields in a blended coherent powerful structure. The model takes a shot at four association crossing points. They created prescient conditions as a paired variable. In a four-stage crossing point at a time one light is green. The request for the lights is and the conditions are given which must fulfill. The yield from the conditions can be blended in the MDL system which predicts the quantity of vehicle draws near. At long last, to foresee the in-stream rate they utilized the ARMA model.

In the diary [5], this model was created to destroy the truck gridlock issue in a specific zone in Bombay. The specialists did a fundamental examination and gathered information. They gathered the day by day information which was changed into the month to month information and afterward they determined the methods and standard deviation of the month to month information. A while later, they did an otherworldly examination. At that point, they shaped normalized information. The chose ARMA and ARIMA model for the issue dependent on MMSE and MLR standards. These two were utilized for the choice of the best model for every hallway. From that point forward, they displayed approval. The reason for the model's approval is to ensure that the presumptions made during the advancement were legitimate. Also, they broke down the qualities structure MMSE, MLR for the up-and-comer models.

The traffic stream estimate is imperative to explore in present-day canny transportation and a precise traffic forecast is a reason and a vital aspect for accomplishing traffic light and arranging. Long haul traffic stream conjectures depend on hours, days, months, and even a long time for the unit of time, which is imperative to the traffic gauge. On one hand, it will help lead street support in the activity division and timetable the development progress opportune. We can see that the combinational model is a powerful method to improve the exact pace of the traffic stream conjecture. To improve the predicting exactness of the traffic stream, this paper

[9] proposes two combinational conjecture models dependent on GM, ARIMA, and GRNN. Initially, it proposes the idea of partner gauge and the weight circulation strategy dependent on equal total rate mistake and afterward utilizes GM, ARIMA, and GRNN to set up a combinational model of interstate traffic stream as indicated by the fixed weight coefficients. At that point, it proposes the utilization of neural organizations to decide variable weight coefficients and builds up the Elman combinational estimate model dependent on GM, ARIMA, and GRNN, which accomplishes the coordination of these three people. To put it plainly, the combinational model can conquer the weaknesses of different autonomous models and improve expectation execution. Regardless of whether the expected consequence of a model isn't ideal, when joined with another great forecast model, it can likewise improve the expectation impact instead of bargain expectation impact.

Organization traffic displaying request calculations that are equipped for managing the self-comparable conduct of traffic information where regular techniques and suppositions miss the mark as far as precision. Two diverse demonstrating approaches, autoregressive coordinated moving normal (ARIMA was applied to display the institutional organization traffic information. For this examination, three distinct cases are considered for displaying with 500, 1,000, and 1,500 examples, separately, of institutional remote organization traffic. We find in paper [7] that ARIMA performs in a way that is better than all different cases. Notwithstanding, this precision is accomplished to the detriment of computational multifaceted nature. The consistency of organization traffic is a huge enthusiasm for some spaces, for example, blockage control, affirmation control, and organization the board. An exact traffic expectation model ought to be able to catch unmistakable traffic attributes, for example, long-range reliance (LRD) and self-comparability in the huge time scale, multifractal on a little league scale. In this paper, we propose another organization traffic forecast model dependent on non-direct time arrangement ARIMA. This model consolidates the straight time arrangement ARIMA model. We give a boundaries assessment system to our proposed ARIMA model. We at that point assess a plan for our models' forecast. We show that our model can catch unmistakable traffic attributes, on a huge time scale as well as on a little league scale.

2.2 Algorithms/ Models

Time series can be trend cycles, cyclic, seasonal, or irregular. These four components are discussed below:

- **Trend:** It is the change in data over a period of time. Underlying rationale, stochastic, random time series character is provided by a trend that can be deterministic.

- **Seasonal:** It is often of known and fixed frequency. A seasonal pattern occurs when seasonal factors affect the time series—seasonal factors like a particular time of the day or year.

- **Cyclic:** It is the fluctuation of data. Cyclic does not have a fixed frequency, unlike seasonal. [21]

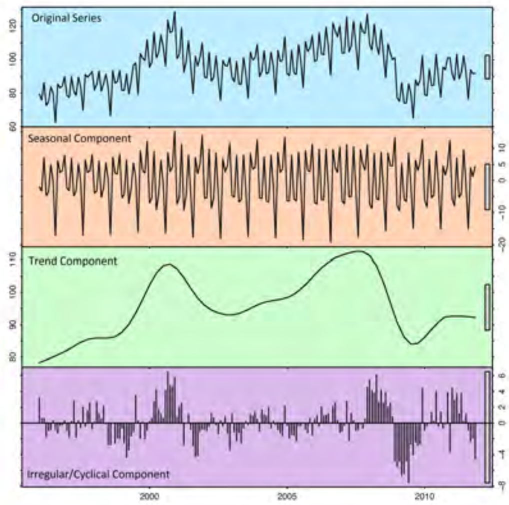

Figure 2.1: Trend, Seasonal, Cyclical and Irregular graphical representation. [6]

- **Irregular:** Data which are represented in a random manner which is due to unknown, unpredictable, non-seasonal and short-term factors for which the future data cannot be predicted here.

Our data exhibits seasonal behaviour, which is of time series data type. The traffic data plotted for each street shows us an increase in slope in a particular pattern, that is, seasonal pattern.

In this work, we present a medium scale comparison study for time series models which would show the highest accuracy in forecasting the traffic data of our seasonal time series dataset. There have been not much comparison studies for time-series forecasting models, hence we decided to do this study which would give us and others more insight on this topic. The models or algorithms considered are ARIMA (Auto-Regressive Integrated Moving Average), SNAIVE (Seasonal Naïve), PROPHET, ETS (Error Trend and Seasonality, or exponential smoothing) and the last one is the combination of all models we named it "mix". The study shows us the significant difference between each of the models.

2.2.1 Autoregressive Integrated Moving Average (ARIMA)

ARIMA deals which various types of time-series data for which it uses its resourceful method which is quite simple but sophisticated in making forecasts for the data. It stands for Auto-Regressive Integrated Moving Average which is a generalisation of the Auto-Regressive Moving Average along with the integration element added. They are a class of statistical algorithms which is used for forecasting and analysing data which are based on time-series. The word ARIMA can divided and explained into:

- AR: It is dependent on the relationship between a few lagged and normal observations. AR stands for Autoregression.

- I: As it is important to convert the time-series into a stationary one, the differencing of raw observations is used. I here stand for Integrated.

- MA: MA uses the proportionality between a residual error from a lagged observation coupled with moving average and a lagged observation. Its full form is Moving Average.

It can be explicitly specified in the model like a parameter in each of these components. Integer values are substituted in place of parameters to point out the accurate model from the ARIMA (p, d, q) and which it is handled quickly. The parameters are:

- p: It is the number of lag observations included in the model. This parameter denotes the amount of lag observations that ARIMA comes with. Hence, it is known as lag order.

- d: This parameter denotes the amount of times that the raw data observations are differenced. Hence, it is known as degree of differencing.

- q: This parameter describes the magnitude of the moving average window. It is also known as the order of moving average.[18]

Error Trend and Seasonality, or exponential smoothing (ETS)

Exponential Smoothing Method handles weighted averages of previous observations to forecast future data. As observations becomes aged (in time), these values' priority gets smaller at an exponential rate (since the current vales are given more importance in the series). They are a family of forecasting models.

ETS connects Error, Trend, and Seasonal components using smoothing techniques. The components can be connected either multiplicatively, additively, or can be removed from the algorithm. Hence, ETS is identified as an ETS framework, wherein a smoothing fashion, the components are calculated.

We can see that, when Error, Trend, and Season are combined, it gives us many combinations in at least three different ways. For example- As we know, the three main components are- Error, Trend, and Season. These three terms is handled using these techniques:

- Additive: A

- Multiplicative: M

- None: N (Not included)

- Additive: Ad

- Multiplicative Damped: Md

Table 2.1: Combination of ETS model [4]

Trend Component	Seasonal Component		
	N(None)	***A(Additive)***	***M(Multiplicative)***
N	NN	NA	NM
A	AN	AA	AM
Ad	AdN	AdA	AdM
M	MN	MA	MM
Md	MdN	MdA	MdM

Therefore, ETS(A,N,N) is ETS with an additive error which is an additive error system with no seasonality nor trend. Keeping the same format in mind, we can say that:

- **ETS(M,N,N):** SES with multiplicative errors.
- **ETS(A,A,N):** Holt's linear method which has additive errors.
- **ETS(M,A,N):** Holt's linear method which has multiplicative errors.[16]

Any specific observation of a time-series is related to its past observations, i.e. the observations preceding it. This approach is built on this fundamental assumption. The models under the exponential smoothing framework include defining an observation in the time-series as a weighted average of its as observations becomes aged (in time), these values' priority gets smaller at an exponential rate.

$$Y_{(t+1)} = \alpha Y_t + \alpha(1-\alpha)Y_{(t-1)} + \alpha(1-)^2 Y_{(t-2)} + \alpha(1-\alpha)^3 Y_{(t-3)} \dots \tag{2.1}$$

A specific observation at time t plus 1 can be stated as weighted average of all the past observations during times t, t-1, t-2, . . . , 2, 1 in this series. Smoothing parameter controls the rate at which the weights decay exponentially which is denoted by, which varies between 0 1, as it is seen in the generic mathematical representation of the idea.

Simple Exponential Smoothing: We start by observing the simplest model in the framework which builds on the above assumption of exponentially decreasing weights. Simple Exponential Smoothing does not take into account any seasonal or trend effects and hence, is perfect for application on time-series with very weak Seasonality Trend.

Defining the Modelling Equation: A generic observation of the time-series can be showcased drawing from the idea as mentioned earlier, is-

$$\hat{Y}_{(t+1)} = \alpha Y_t + \alpha(1-\alpha)Y_{(t-1)} + \alpha(1-)^2 Y_{(t-2)} + \alpha(1-\alpha)^3 Y_{(t-3)} \dots \tag{2.2}$$

- The notation $\hat{Y}_n$ refers to the estimated value at time **n**
- The notation Y_n refers to the actual value at time **n**

The notation can be simplified further into;

$$\hat{Y}_{(t+1)} = \alpha Y_t + \alpha(1-\alpha)Y_{(t-1)} + \alpha(1-)^2 Y_{(t-2)} + \alpha(1-\alpha)^3 Y_{(t-3)} \dots \tag{2.3}$$

$$\hat{Y}_{(t)} = \alpha Y_{t-1} + \alpha(1-\alpha)Y_{(t-2)} + \alpha(1-)^2 Y_{(t-3)} + \alpha(1-\alpha)^3 Y_{(t-4)} \dots \tag{2.4}$$

Comparing the above two equations, we can simplify $\hat{Y}_{(t+1)}$ as

$$\hat{Y}_{(t+1)} = \alpha Y_t + (1-\alpha)\hat{Y}_{(t)} \tag{2.5}$$

Computing the model parameters
Every model under ETS framework requires an initialisation of the first observation of the series, in order to provide closure for the series.

$$\hat{Y}_{(t+1)} = \alpha Y_t + (1-\alpha)\hat{Y}_{(t)} \tag{2.6}$$

The assignment, as mentioned above, is not the only option for initialising the model. We could also set the value to the average or mean of the series, among other alternatives. The next stage involves the calculation of the parameter, , once initialised. We follow a suitable way similar to that of in Multiple Linear Regression to choose the optimum value for the parameter. We choose a loss function as the Sum of Squared Errors (SSE) and

$$SSE = \sum (Y_n - \hat{Y}_n)^2 \tag{2.7}$$

select the optimum value of the parameter, , which minimises the loss function now.

Extending the Simple ETS Model to adjust for Trend and Seasonality
In most of the series under consideration, fully understanding the stages in the computation of the model provides us with a stepping stone for the more complex models in the framework, though the use-cases of the Simple ETS model are limited due to the presence of a Seasonal or Trend component. The Seasonality Trend attributes are characterised explicitly through separate equations which are characterised by their independent parameters, to expand the scope of the Simple ETS model to time-series with these components. [12]

2.2.2 Seasonal Naïve (SNAIVE)

Sometimes the data available is not enough and data which is of time-series type. In such circumstances, the naïve method is used which uses previous data from the last observation to make the prediction for the next data. Now Seasonal naïve (SNAIVE) works for data which are very seasonal in nature. SNAIVE makes prediction in the same way as naïve month expect for it predicts the last observed data from the same season of the year.[15] Now let us look at the equations which help us do these forecasting in a straightforward way. The recent observation is set as the value of all forecast, in case of dealing with naïve forecasts, which is,

$$\hat{Y}_{T+h=}\hat{Y}_T \tag{2.8}$$

A naïve forecast is called a random walk forecast as the optimal data here follows a random walk.

When the dataset is seasonal, we use SNAIVE forecast. Here, the recent observation is set as the value of all forecast and the last observation here has to be from same season from any year, for example January of the recent year. Typically, for T+h (time), the forecast equation can be represented as,

$$\hat{Y}_{T+h=}\hat{Y}_{T+h-k}m \tag{2.9}$$

where = seasonal period and $k = ((h-1))/m + 1$ where k is the integral part of $(h-1)/m$ which tells the total number of years before **T+h** (time) in the dataset. For any sets of data, the all forecast of a particular future time t will be equal to the very last observed data of the time t but of different year or season. [23]

2.2.3 Prophet Model

The Prophet is an open-source library, developed by Facebook, which is developed for building forecasts for univariate (single variable) time-series datasets. It is also known as Facebook Prophet which was published by Facebook's Core Data Science team. It is very easy to use. It is built in such a way that it can make accurate forecasts for data having seasonal and trend behaviour. This model is a perfect match for time-series data containing having rich seasonal traits. Also, seasonal historical data works like a charm for this model.

Prophet model supports three components, which are, trends, seasonality and holidays. Now, using these three components what this model does is implement another model named additive time-series forecasting model. This model library can be handled using Python and R language. It is very easy to apply as it is completely automatic.[19]
The procedure makes use of a decomposable time-series model with three main model components: trend, seasonality, and holidays. As we discussed earlier, this model uses an additive time-series forecasting model using its three components.

It is as same as the Generalised Additive Model (GAM). Hence it uses many of the GAM's components in its equation to carry out its forecasting with time as a regressor. In its simplified form, the equation will be-

$$\mathbf{y(t) = g(t) + s(t) + h(t) + e(t)} \tag{2.10}$$

From which,

- $\mathbf{g(t)}$: The trend models changes which are non-periodic.

- **s(t)**: The seasonality deals with changes which are periodic, that is, weekly, monthly or yearly.
- **h(t)**: The holiday model deals with potential irregular schedules.
- **e(t)**: It deals with peculiar changes which are not handled by Prophet.

Therefore, the procedure's equation is,

$$y(t) = piecewise_t rend(t) + seasonality(t) + holiday\ effects(t) + i.i.d.\ noise$$

In case of the trend model for g(t), the equation provides two different models- one is the peicewise trend linear model and the other is the saturating growth model.

When the prophet model is extrapolated past the history to make an accurate forecast, g(t) is supposed to go on a constant rate. The forecast trend makes an estimate of the uncertainty. This is found out by extrapolating forward the generative model.

The seasonal component, s(t), allows periodic changes in case of days, weeks or years (daily, weekly or yearly). This is what gives the model its adaptability nature.

The effect of specific holiday in most cases is almost the same every year, which makes it an essential addition to the forecast and also part of it. The h(t) component deals with predictable events or happenings of the year, along the ones with the irregular schedules.

The user is required to give a customized event's list, to utilise the feature. Fusing this holiday's list in the above equation is created very easily by taking the holiday's effect to independent. [20]

2.2.4 Mix Model

Statistical models have been very commonly been used in time-series forecasting and data analysis. But the model selected out of the many statical models, is many times not stable. For this reason, the forecasted value we get from using these models can be very different from the actual value. [3] There is a high chance that a single forecasting technique or model would not be of any use in all scenarios. Each of the forecasting methods comes with their own unique feature and can be used based on their different approaches. we came to a decision to combine the models we used in this paper and create a model composed all the models we used. [10] The models or algorithms we have used in this thesis- ARIMA (Auto-Regressive Integrated Moving Average), SNAIVE (Seasonal Naïve), PROPHET and ETS (Error Trend and Seasonality). Hence, we decided to combine all these models into one forecasting model and use a model capable of providing a reliable and accurate forecast

Chapter 3

Proposed Model and Implementation

3.1 Dataset Description

We have used a dataset from a project called Future Melbourne 2026 Plan. This data set was released as a public resource to use. A contractor was hired by The City of Melbourne to conduct the traffic counts on roads through different roads in the city. The vehicles were categorized into 12 categories and recorded count on per hour.

```
## Rows: 60,168
## Columns: 29
## $ date                     <chr> "03-09-15", "03-09-15", "03-09-15", "03-09-...
## $ road_name                <chr> "Bayles Street", "Bayles Street", "Bayles S...
## $ location                 <chr> "Between Fitzgibbon Street and Jageurs Lane...
## $ suburb                   <chr> "Parkville", "Parkville", "Parkville", "Par...
## $ speed_limit              <dbl> 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40,...
## $ direction                <chr> "E", "E", "E", "E", "E", "E", "E", "E", "E"...
## $ time                     <time> 00:00:00, 01:00:00, 02:00:00, 03:00:00, 04...
## $ vehicle_class_1          <dbl> 0, 0, 0, 0, 1, 1, 4, 14, 16, 25, 23, 31, 22...
## $ vehicle_class_2          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ vehicle_class_3          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 2, 2...
## $ vehicle_class_4          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0...
## $ vehicle_class_5          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ vehicle_class_6          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ vehicle_class_7          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ vehicle_class_8          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ vehicle_class_9          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ vehicle_class_10         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ vehicle_class_11         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ vehicle_class_12         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ vehicle_class_13         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ motorcycle               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0...
## $ bike                     <dbl> 0, 0, 0, 0, 1, 0, 0, 5, 0, 2, 1, 1, 1, 0, 0...
## $ `Total Vehicles`         <dbl> 0, 0, 0, 0, 2, 1, 4, 19, 17, 28, 24, 34, 24...
## $ average_speed            <dbl> 0.0, 0.0, 0.0, 0.0, 18.2, 44.7, 27.4, 27.3,...
## $ `85th_percentile_speed`  <dbl> 0.0, 0.0, 0.0, 0.0, 18.0, 44.0, 32.0, 38.0,...
## $ maximum_speed            <chr> "-", "-", "-", "-", "18", "45", "32", "41",...
## $ road_segment             <dbl> 22353, 22353, 22353, 22353, 22353, 22353, 2...
## $ road_segment_1           <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ road_segment_2           <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
```

Figure 3.1: Overview of the dataset

Without any processing and preparation, we can see that we have 60,168 observations and 29 columns. Date, Time, Total vehicles, and road names are the target variables for this forecasting. More on this is going to be explained in the data preprocessing part.

3.1.1 Data preprocessing

The independent variable is time, for this time-series analysis. In the dataset, we have date and time in a separate column, and they are of type 'character'. We need to concatenate these two columns and convert the type to 'timestamp,' and we identify 'Total Vehicles' as the target variable.

```
dataset %>%
  distinct(road_name) %>%  # get distinct road name
  arrange(road_name) %>%  # order by names then display
  reactable(highlight = TRUE,defaultPageSize = 5, # display the dable
            defaultColDef = colDef( headerStyle = list(background = "#4285F4")))
```

road_name
Abbotsford Street
Abbottsford Street
Adderley Street
Agnes Street
Anderson Street

Figure 3.2: Anomalies in the dataset

As we can see, this column contains 111 road names, but we notice that some names are implicitly duplicated because of abbreviations and writing errors: for example, Abbotsford Street vs. Abbottsford Street, Domain St vs. Domain Street.

In total, we have identified four errors that we will correct manually; in the most significant dataset, we can use the Levenshtein distance or advanced NLP techniques that can be used to do this work. We build our target variable y as the aggregation of road traffic per road and hour.

```
dataset %>%
  mutate(road_name= case_when(
    road_name%in%c("Abbotsford Street", "Abbottsford Street")~"Abbottsford Street",  # equivalent if then
    road_name%in%c("Domain St", "Domain Street")~"Domain Street", # equivalent if then
    road_name%in%c("Stubbs street", "Stubbs Street")~"Stubbs Street", # equivalent if then
    road_name%in%c("Wells St", "Walsh Street")~"Walsh Street", # equivalent if then
    T~road_name # in all other cases render the correspanding road name
  ) ) %>%
  mutate(date= dmy_hms(paste0(date, time)  )) %>% # create timestamp column using the function dmy_hms from lubridate and  paste0 allow concatenation of strings
  select(road_name, date, `Total Vehicles`) %>%  # selecting important columns
  group_by(road_name, date) %>%
  summarise(y= sum(`Total Vehicles`,na.rm = T )) %>% # calculate target variable by road and date
  ungroup() %>%  # we ungroup the dataframe then group by road name
  group_by(road_name) %>%
  summarise(
    min_date = min(date), # minimum date by road
    max_date = max(date), # maximum date by road
    n_rows = n() # total number of records
  ) %>%
  mutate(expected_rows =as.integer(difftime(max_date,min_date, units = "hours"))+1) %>% #expect number of hours
  filter(expected_rows==n_rows) -> clean_road # get only roads with complete data  and save the result in the dataframe clean_road
```

Figure 3.3: Code snippet of the data cleaning process

As we can see on the different graphs, the seasonality is very strong, and all the series seem stationary.

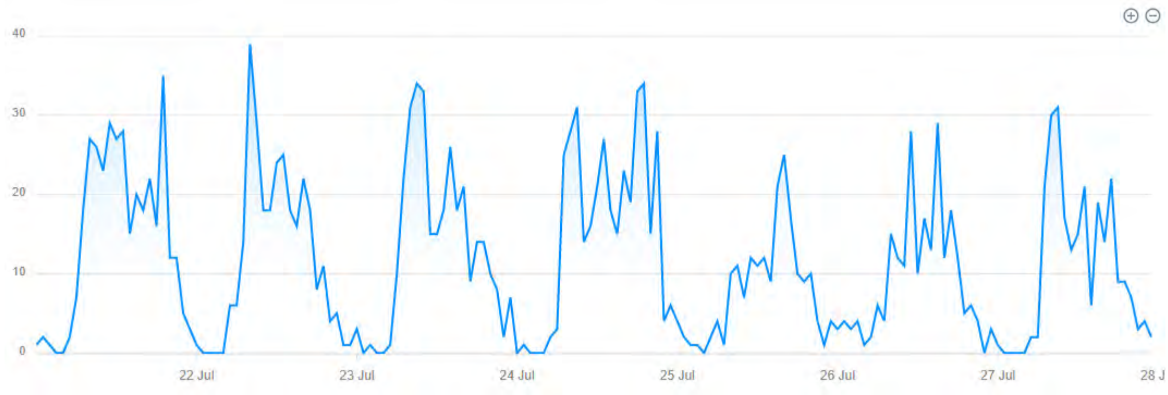

Figure 3.4: The graph of the first street after preprocessing and cleaning the data

3.1.2 Feature selection

In the dataset, the target variables are road name, time, date, total vehicles, and the rest of the columns are not much relevant. In the data description section, we have seen that there are some unnecessary data in the dataset. For instance, there are different vehicle classification types in a different column, description, road segments, and some vehicle classification columns that have missing values. We can drop such columns to improve the training data. [Fig. 3.3]

There are some redundant roads due to typing mistakes and abbreviations. For instance, Abbotsford Street vs. Abbottsford street and Domain St. vs. Domain Street. There are four total errors in total. We need to handle these errors to improve data integrity.

3.2 Implementation

3.2.1 Programming tools and IDEs

The forecasting models are implemented using the R programming language in RStudio. We have used liberties like lubrdate, xts, tibble, fable, fabletools, fable.prophet, feasts, fasster to implement the models and rmdformats, dygraphics, apexcharter, etc. for formatting, plotting graphs, charts.

3.2.2 Target variables for Time Series

As we know, there must be a time variable, based on which the time series models are implemented, and the should be a target variable that would be predicted eventually. In the data preprocessing, we discussed how we altered hour data, which means merging date and time column into one and converted the data type from 'char' to 'timestamp'. Furthermore, we choose 'Total Vehicles' as the target variable since we want to predict the number of vehicles. Finally, we removed the anomalies in road names and used the *'roadname'* as a unique identifier. [Fig. 3.3]

3.2.3 Plotting after cleaning the data

After cleaning up the dataset as per our requirement, we plotted the 'Total Vehicles' variable with time where time is the independent variable [Fig-3]. The figure shows that the seasonality is very strong, and all the series seem stationary. Almost all the streets have seasonality, and it is quite familiar with traffic data since the amount of traffic passing through a street depends a lot on time.

3.2.4 Forecasting models

We have over a hundred unique roads. Though most of them have seasonal data, some of them have unpredictable data too. That is why we want to implement four different forecasting models to see which one performs better for which kind of data. We have chosen four different prevalent and advanced forecasting models: 'ARIMA', 'ETS', 'SNAIVE' and 'PROPHET' and lastly, we have a mixed model that gives us the average of all four models. For implementing these models, we have used two frameworks, namely fable, and prophet. The fable package is capable of handling many time series all at once.

3.2.5 Implementing models using fable package

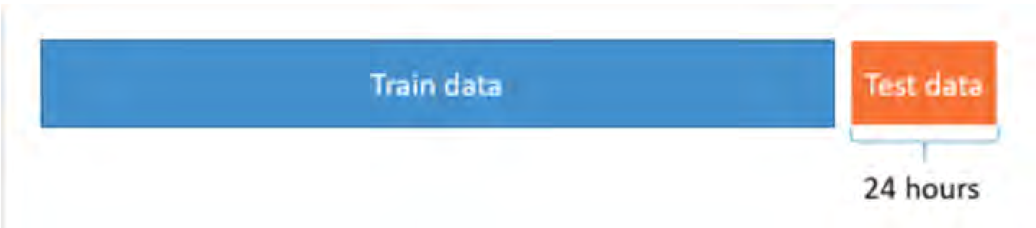

Figure 3.5: Training data and test and forecast horizon

From our primary dataset, we set the entire data as training data except for the last 24 hours since we want to set the forecast horizon as 24 hours, and the time step the data has is an hour. So the last 24 hours historical data is our test data.

Even though the fable package is capable of handling several time series all at once, we decided to use on a single time series. We have fitted all the models in a single call.

```
Agnes %>%
  model(
    arima = ARIMA(y), # fit best ARIMA model using auto.arima
    ets = ETS(y) , # fit the best ETS model
    snaive = SNAIVE(y), # fit seasonal naive model
    prophet = prophet(y~ season("day")+season("week")+season("year")) # fit prophet with multiple seasons
  ) %>%
  mutate(
    mixed = (ets + arima  + snaive +prophet) / 4
  )-> models
```

Figure 3.6: Fitting all models

The object returned is known as a ‘mable’ which is a model table. Here each of the cells directly corresponds to a fitted model. Our mable is only one row long as only one time-series is used against our fitted models.

To forecast using all the models on our time-series dataset, we then send the object to the ‘forecast‘ function with the horizon of 24 hours.

```
```{r eval=FALSE}
fc <- models %>%
  forecast(h = 24)
```



```{r}
print(fc)
```
```

Figure 3.7: Forecasting with a horizon of 24 hr

The object returned is known as a ‘mable’ which is a model table which consists of:

- - the .model column becomes an additional key;
- - the y column contains the estimated probability distribution of the response variable in future time-periods;
- - the .mean column contains the point forecasts equal to the mean of the probability distribution.

3.2.6 Plotting the values using dygraphs

Figure 3.8: Plot of forecast on Agnes Street

3.2.7 Forecast error calculation

To compare these models' forecast accuracy, we will make a new dataset from the original one for training which not contains traffic data for the last 24 hours. We aim to forecast the last 24 hours, which we have not trained, and then compare those forecasted values with the actual ones. [Fig. 3.5]

A forecast "error" can be defined as the difference in the forecasted value compared to the actual one. We can measure forecast accuracy by using various techniques related to forecasting errors.

The three most commonly used and famous forecasting error measures we going to use are: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE).

Among the three forecasting error measuring formulas, MAE is very easy to calculate and understand. Minimizing MAE subsequently directs towards the forecast of median whereas for RMSE it directs towards the forecast of the mean. Consequently, RMSE is used quire more often even though it is more challenging to understand. The best model has to be the one that minimizes the MAE, RMSE, or MAPE.

Percentage errors are easier to interpret as a percentage is a unit-free value which can be depicted out of 100. It is quite advantageous when it comes to comparing different forecasting model's performance on a given time-series dataset.

There is a problem which comes while dealing with percentages. If the actual value or observed value is zero or close to zero then you get your MAPE to be undefined. Also, greater than 100 percent can also happen. Hence it very problematic in many cases to handle MAPE.

````
```{r eval=F}
fabletools::accuracy(fc_train, sdf) %>% select( .model, road_name, .type, RMSE, MAE, MAPE)  -> res
```

You can filter road and get the metrics by typing the name in `search`

```{r}
reactable(
  res %>% mutate(MAE= round(MAE, 2),RMSE= round(RMSE, 2)) %>% arrange(road_name),
  defaultColDef = colDef( headerStyle = list(background = "#4285F4")),
  searchable = T,defaultPageSize = 5,highlight = T)
```
````

Figure 3.9: Code snippet of error calculation

.model	road_name	.type	RMSE	MAE	MAPE
arima	Agnes Street	Test	6.22	4.14	Inf
ets	Agnes Street	Test	5.9	4.39	Inf
mixed	Agnes Street	Test	5.9	3.83	Inf
prophet	Agnes Street	Test	5.17	4.06	Inf
snaive	Agnes Street	Test	8.38	5.84	Inf

1–5 of 385 rows Previous 1 2 3 4 5 ... 77 Next

Figure 3.10: Error calculation table

3.3 System Implementation

For developing a live system, we have used R markdown(rmdformats). R markdown is popular because it can be used to mark down and write R code chunks in between, and most notably, with the help of libraries, it becomes a lot powerful. For example, we used the 'knitr' library to develop dynamic report generation, 'ractables' for interactive data tables, 'shiny' for developing interactive web apps, which we eventually deployed to ShinyApps.

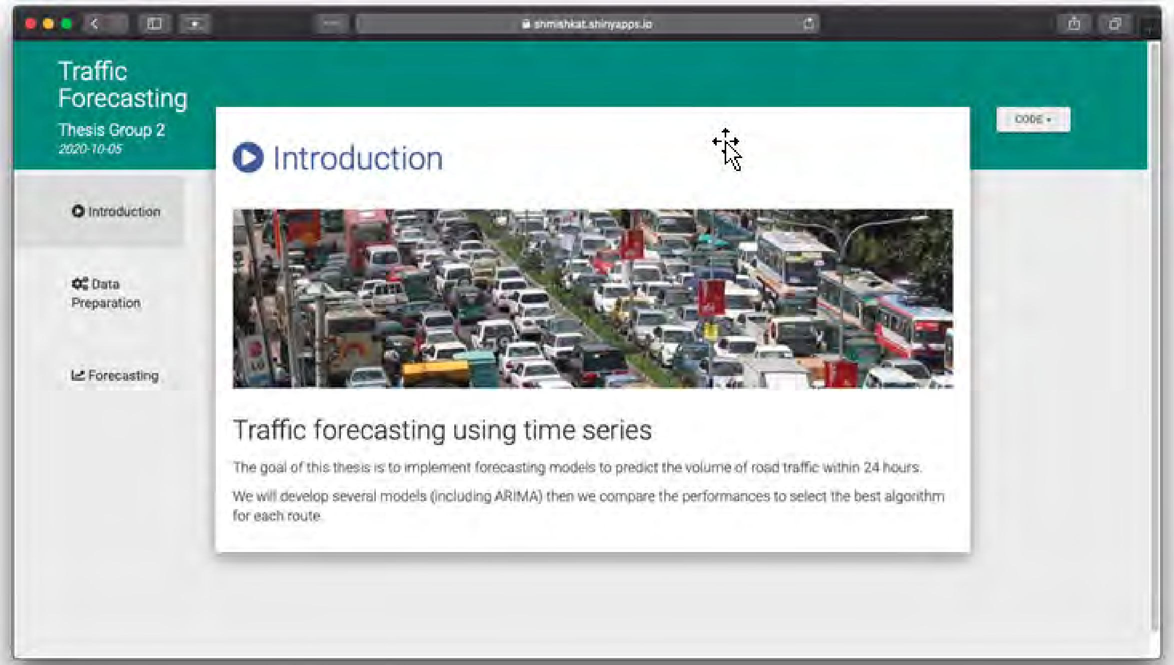

Figure 3.11: Snapshot of the implemented system (ShinyApp)

Chapter 4

Result Analysis

4.1 Root mean squared error (RMSE)

RMSE is stands for root mean squared error and it is a quadratic scoring decision that measures the average magnitude of error. It is the square root of the average of squared differences between prediction and actual observation.

$$RMSE = \sqrt{1/n \sum_{j=1}^{n} (y_j - \hat{y}_j)^2} \tag{4.1}$$

Root-mean-square error (RMSE), also called the root-mean-square deviation (RMSD), and is regularly used to a proportion of the contrasts between values anticipated by a model or an assessor and the values observed. This speaks to the square base of the second example snapshot of the contrasts between the watched qualities or the quadratic mean of these distinctions and anticipated qualities. These deviations are known as residuals when the computations are performed over the information test utilized for assessment and are called errors when determined out-of-test. The RSME serves to total the extents of the mistakes in forecasts for different occasions into a solitary prescient force measure. It is a proportion of precision, to think about gauging blunders of various models for a specific dataset and not between datasets, as it is scale-subordinate.

RSME is consistently non-negative, and an estimation of 0 would demonstrate a 100% ideal fit for the information. Normally, a lower RMSD is superior to a higher one. Be that as it may, examinations across various information types would be invalid because the measure relies upon the size of the numbers utilized.

RSME is the square root of the average of squared errors. The impact of every error on RMSD is proportional to the squared error; in this way, bigger errors have a lopsidedly enormous impact on RMSD. Subsequently, RMSD is delicate to anomalies.

4.2 Mean Absolute Error (MAE)

Mean AbsoluteError (MAE) is an estimation of errors between paired perceptions which are communicating a similar wonder, in statistics. Instances of Y versus X incorporate one estimation method versus an elective procedure of estimation and correlations of anticipated versus observed, subsequent time versus initial time. MAE is determined as:

$$MAE = \sum_{i=1}^{n} |y_i - x_i|/n = \sum_{i=1}^{n} |e_i|/n \tag{4.2}$$

Therefore it is an arithmetic average of the absolute errors $|e_i| = |y_i - x_i|$, where y_i is the prediction and x_i the true value. MAE utilizes the very same scale as the information being estimated. This is called as a scale-subordinate precision measure and subsequently can't be utilized to make examinations between arrangements utilizing various scales. The mean absolute error is a common measure of forecast error in time series analysis.

4.3 Mean Absolute Percentage Error (MAPE)

Mean absolute percentage error (MAPE), which is also known as mean absolute percentage deviation (MAPD), is a measure of prediction accuracy of a forecasting method in statistics. For instance, in pattern assessment, it is additionally utilized as a misfortune work for relapse issues in AI. It for the most part communicates the exactness as a proportion characterized by the equation:

$$M = 1/n \sum_{t=1}^{n} |A_t - F_t|/A_t \tag{4.3}$$

Where Ft is the forecast value and At is the actual value. The MAPE is additionally here and there announced as a rate, which is the above condition increased by 100. The distinction between Ft and At is divided by the Actual value At again.This present figuring's total worth is added for each estimated point in time and partitioned by the quantity of fitted points n. Multiplying it by 100% makes it a percentage error.

RSME gives a misfortune capacity of the subsequent degree, and its proportion of vulnerability is anticipating. It has the advantage of punishing enormous errors all the more so it very well may be more suitable now and again, for instance, if being off by ten is more than twice as awful as being off by 5. One particular favorable position of RMSE over MAE is that RMSE stays away from the utilization of taking the total worth, which is bothersome in numerous numerical calculations. Mean absolute percentage error (MAPE) is a percentage, so we can without much of a stretch think about it among arrangement, and individuals can undoubtedly comprehend and decipher rates. As a rate, MAPE just bodes well for values where proportions and divisions bode well. It looks bad to ascertain rates of temperatures, for example, so you ought not to utilize the MAPE to compute the precision of a temperature estimate. On the off chance that simply a solitary real is zero, At=0, at that point you separate by zero in figuring the MAPE, which is unclear. More noteworthy than 100% can happen in MAPE. Presently if you want to work with exactness, which a few people characterize as 100%-MAPE, at that point, this may prompt pessimistic precision, which individuals may struggle to understand.

In this way, we can say that RSME is a superior decision over MAPE as MAPE causes a lot of befuddling during the investigation of information. Since RMSE is mostly used for forecasting error evaluation, we have chosen RMSE for our result analysis.

model	road_name	type	RMSE	MAE	MAPE
arima	Agnes Street	Test	6.22	4.14	Inf
ets	Agnes Street	Test	5.9	4.39	Inf
mixed	Agnes Street	Test	5.9	3.83	Inf
prophet	Agnes Street	Test	5.17	4.06	Inf
snaive	Agnes Street	Test	8.38	5.84	Inf

1–5 of 385 rows Previous 1 2 3 4 5 ... 77 Next

Figure 4.1: Execution of the models regarding RMSE (the lower RMSE, the better the model)

There are 76 streets in the dataset. We have calculated the best performing models and 2nd best performing model in terms of RMSE.

Table 4.1: The best model (Prophet)

Model name	**Prophet**	**Snaive**	**ETS**	**ARIMA**	**Mixed**
No. of occurrence	42	12	12	2	8

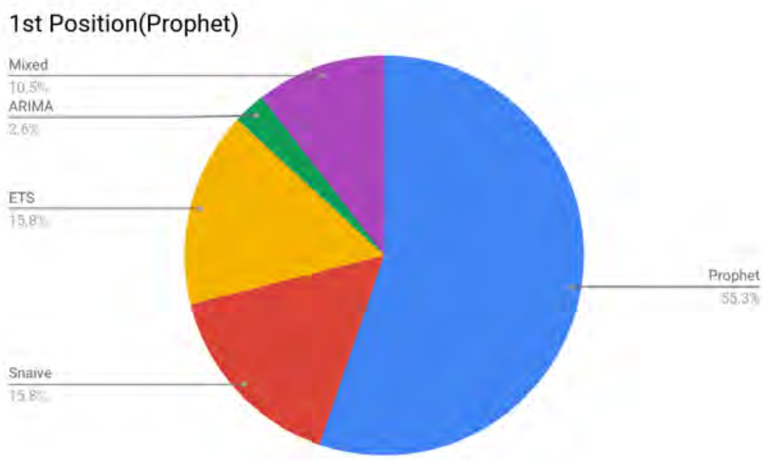

Figure 4.2: Pie chart of the best performing model

Table 4.2: Second best model (Mixed)

Model name	Mixed	Prophet	ARIMA	ETS	Snaive
No. of occurrence	45	11	10	9	1

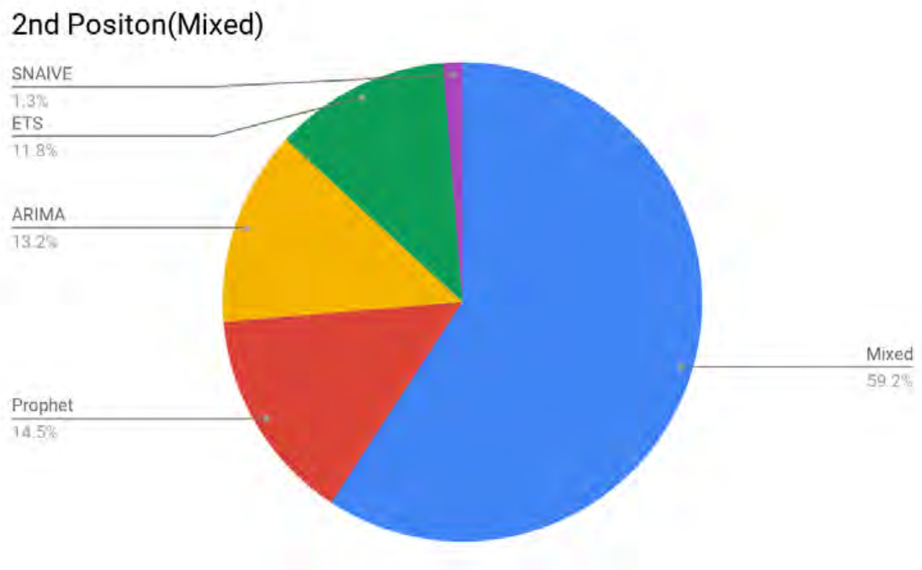

Figure 4.3: Pie chart of the second-best performing model.

4.4 Data with less seasonality:

There are some streets where the data is less seasonal. We have observed that the models behave differently with less seasonal data. As we can see in the table below that in less seasonal data, the performance of the model 'Prophet' is the worst on the contrary, in seasonal data 'Prophet' is the clear winner but in our data set only 9 roads to have less seasonal data rest of the roads have strong seasonality.

Table 4.3: Less seasonality(SNAIVE)

Model name	Snaive	ETS	Mixed	ARIMA	Prophet
No. of occurrence	4	2	2	1	0

Out of 76 roads, 67 roads have strong seasonality in the data and the other 9 roads have less seasonality. 'Prophet' is the best model for the streets with strong seasonal data and Snaive is the best model for the roads with less seasonal data. Since the majority of the roads have seasonal data in this dataset and traffic flow on a particular road usually has strong seasonality we can say that 'Prophet' the best performing model.

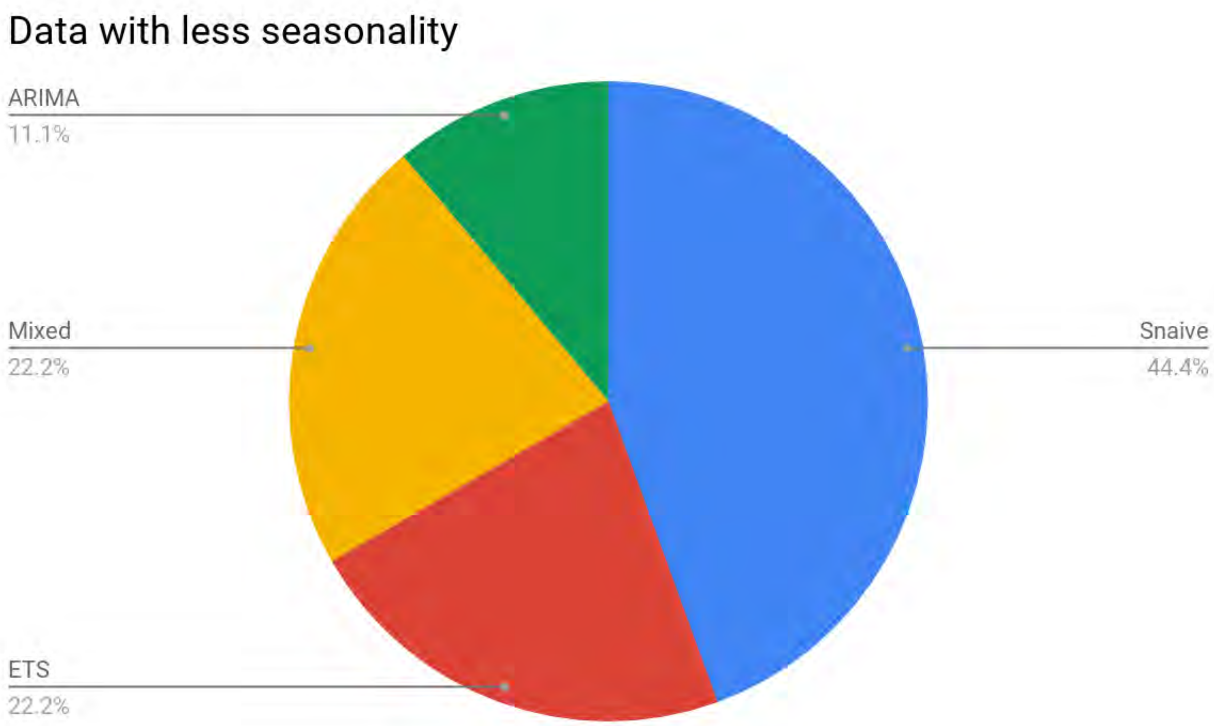

Figure 4.4: Pie chart of best performing model in less seasonal data (Snaive).

Chapter 5

Conclusion and Future Work

5.1 Conclusion

We wanted to find the most suitable forecasting model based on time-series which helps us to forecast future traffic data when there is enough dataset is provided. When this goal in mind, we began to search for models based on prediction, which would enable us to predict the value data. However, upon more research, we found that it, not a prediction but rather forecasting, after which we focused on that. We came across so many time-series forecasting models that it made our work both tedious and fun at the same time. While working with these forecasting models, we understood that not all the models can prompt a precise forecast. The models or algorithms we decided on and focused on the end were ARIMA (Auto-Regressive Integrated Moving Average), ETS (Error Trend and Seasonality/exponential smoothing), SNAIVE (Seasonal Naïve), PROPHET and the last one is the combination of all models we named it "mix". We examined these diverse models for our goal. While working with these determining models, we comprehended that not all the models can provoke an exact conjecture this investigation had indicated to us the critical contrast between every one of the models and how they act on a given dataset. We forecasted the daily amount of traffic flow on the major routes in Australia, finding the total of vehicles for each major street by testing and training each model to check for the accurate one.

After analyzing the performance of these algorithms, we used in our project. We were able to find out the model with better accuracy with our dataset with an accuracy of 90% compared to the other models that we have used. That model is PROPHET.

5.2 Future Work

We want to integrate our model with Google maps so that the user would have no problem in understand the traffic situation of a particular route beforehand. This model will be a mixture of the best time-series forecasting models that we find. As a solitary determining procedure or model cannot work in every single circumstance. Every one of the estimating strategies has its particular use case and can be applied concerning numerous elements and the yield required as various methodologies had their interesting qualities and shortcomings, we need to consolidate more techniques and attempt to make the most refined and productive model which give exact future readings near great. We want to make our user interface more user friendly for the users. We want to increase our forecasting time more.

5.3 Feedback From Panel

We were asked the reason behind not using SARIMA (Seasonal Autoregressive Integrated Moving Average) since our dataset was based on seasonal data. Our reasoning was that, we used ARIMA to work with time-series dataset that are not just seasonal but also non-seasonal, since traffic data not always display seasonality in all cases.

Bibliography

[1] F. I. Hira, M. F. Maruf, and A. Hossain, *Stock market prediction using time series analysis: Semantic scholar*, Jan. 1970. [Online]. Available: https://www.semanticscholar.org/paper/Stock-market-prediction-using-time-series-analysis-Hira-Maruf/ee579a52472ddd2ed37c663630dd82ba490aff57.

[2] M. Karimpour, A. Karimpour, K. Kompany, and A. Karimpour, *Online traffic prediction using time series: A case study*, Jan. 1970. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-59387-6_15.

[3] H. Zou and Y. Yang, "Combining time series models for forecasting," *International Journal of Forecasting*, vol. 20, no. 1, pp. 69–84, 2004. DOI: 10.1016/s0169-2070(03)00004-9.

[4] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: The forecast package for r," *Journal of Statistical Software*, vol. 27, no. 3, Jul. 2008. DOI: 10.18637/jss.v000.i00. [Online]. Available: https://cran.r-project.org/web/packages/forecast/vignettes/JSS2008.pdf.

[5] S. L. Dhingra, P. P. Mujumdar, and R. H. Gajjar, *Application of time series techniques for forecasting truck traffic attracted by the bombay metropolitan region*, Jan. 2010. [Online]. Available: https://onlinelibrary.wiley.com/doi/pdf/10.1002/atr.5670270303.

[6] LIBESA, *Using decomposition to improve time series prediction quantdare*, Sep. 2014. [Online]. Available: https://quantdare.com/decomposition-to-improve-time-series-prediction/.

[7] R. K. Yadav and M. Balakrishnan, "Comparative evaluation of arima and anfis for modeling of wireless network traffic time series," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, 2014. DOI: 10.1007/13638.1687-1499.

[8] S. Jiang, S. Wang, Z. Li, W. Guo, and X. Pei, "Fluctuation similarity modeling for traffic flow time series: A clustering approach," *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015. DOI: 10.1109/itsc.2015.143.

[9] Z. Yu, T. Sun, H. Sun, and F. Yang, "Research on combinational forecast models for the traffic flow," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–10, 2015. DOI: 10.1155/2015/201686.

[10] T. Firman, *Combining different methods to create advanced time series prediction*, Nov. 2016. [Online]. Available: https://www.kdnuggets.com/2016/11/combining-different-methods-create-advanced-time-series-prediction.html.

[11] X. Yu, S. Xiong, Y. He, W. E. Wong, and Y. Zhao, *Research on campus traffic congestion detection using bp neural network and markov model*, Sep. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212616301673.

[12] Admin, *Exponential smoothing framework: Big data: Predictive analytics*, Oct. 2017. [Online]. Available: https://www.latentview.com/idealab/exponential-smoothing-ets-framework/.

[13] S. V. K. G. Omkar, *Time series decomposition model for traffic flow ...* 2017. [Online]. Available: https://www.researchgate.net/publication/325521808_Time_series_decomposition_model_for_traffic_flow_forecasting_in_urban_midblock_sections.

[14] A. A. Haider, *Traffic jam: The ugly side of dhaka's development*, May 2018. [Online]. Available: https://www.thedailystar.net/opinion/society/traffic-jam-the-ugly-side-dhakas-development-1575355.

[15] Sangarshanan, *Time series forecasting - smoothing models*, Sep. 2018. [Online]. Available: https://medium.com/@sangarshananveera/time-series-forecasting-part-1-d523bcc3acbf.

[16] Daitan, *Exponential smoothing methods for time series forecasting*, Aug. 2019. [Online]. Available: https://medium.com/better-programming/exponential-smoothing-methods-for-time-series-forecasting-d571005cdf80.

[17] X. Zhang, H. Liu, Y. Zhao, and X. Zhang, *Multifractal detrended fluctuation analysis on air traffic flow time series: A single airport case*, Jun. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437119310362.

[18] J. Brownlee, *How to create an arima model for time series forecasting in python*, Aug. 2020. [Online]. Available: https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/.

[19] ——, *Time series forecasting with prophet in python*, May 2020. [Online]. Available: https://machinelearningmastery.com/time-series-forecasting-with-prophet-in-python/.

[20] W. Robson, *The math of prophet*, Oct. 2020. [Online]. Available: https://medium.com/future-vision/the-math-of-prophet-46864fa9c55a.

[21] V. Saratchandran, *Why time series forecasting is a crucial part of machine learning: Fingent blog*, Jan. 2020. [Online]. Available: https://www.fingent.com/blog/why-time-series-forecasting-is-a-crucial-part-of-machine-learning/.

[22] A. Tuovila, *Forecasting definition*, Sep. 2020. [Online]. Available: https://www.investopedia.com/terms/f/forecasting.asp.

[23] R. J. Hyndman and G. Athanasopoulos, "The forecaster's toolbox, some simple forecasting methods," in *Forecasting: Principles and Practice*, 2nd ed. OText.